UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,296	11/25/2003	Alnasir Ladha	11312	9107

26890 7590 03/31/2009
JAMES M. STOVER
TERADATA CORPORATION
2835 MIAMI VILLAGE DRIVE
MIAMISBURG, OH 45342

EXAMINER	
KUDDUS, DANIEL A	
ART UNIT	PAPER NUMBER
2164	
MAIL DATE	DELIVERY MODE
03/31/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/722,296	**Applicant(s)** LADHA ET AL.	
	Examiner DANIEL KUDDUS	**Art Unit** 2164	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _03_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _25 February 2009_.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _8-25_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _8-25_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. This Office action has been issued in response to amendment filed February 25, 2009. In response to last office action, claims 8, 15, 20, 21 and 22 have been amended. Accordingly, claims 8-25 remain pending in this application. Applicant's arguments are carefully and respectfully considered and some are persuasive, while others are not. Accordingly rejections have been removed where arguments were persuasive, but rejections have been maintained where arguments were not persuasive. Also, a new rejections based on the newly added claims have been set forth. Accordingly, claims 8-25 are rejected and this action has been made **FINAL**, as necessitated by amendment.

Response to Arguments

2. Applicant's arguments presented on February 25, 2009 in response to the office action mailed on November 25, 2008 have been carefully and respectfully considered, but they are not persuasive.

With respect to applicant's argument "*the combined references do not show multiple instances of the same application that divide up work processing to perform parallel processing and work against query results… the combined references do not show or suggest that data is streamed as the query results are produced to application queues and then streamed to application instances. Still further, application data produced is not streamed to load queues. Moreover, all this work results in merged tables once all instances of the application have finished producing application data from the query results. These elements are not shown in any* *fashion in the proposed combination of references"*. The arguments however are not persuasive. The limitations of multiple instances of the same application that divide up work processing to perform parallel processing are not recited in the claims. Klein in fact teaches amended claim recites limitations. Klein teaches the limitation of producing the results that are then streamed to plurality of application queues residing on a plurality of the processing nodes as the results are acquired (see col. 6, lines 1-10, data flows between the nodes of the execution tree are handled by the use of a pair of queues, between parent and child nodes. In particular each parent node is coupled to a child node by a request queue and a fetched records queue. The request queue stores requests being conveyed from the parent node to its child node, while the fetched records queue conveys data and return codes (e.g., an end of file or end of scan code) being returned to the parent node in response to the requests, column 15, line 39-60, column 10, line 44-50, 'streaming queries' may use commands such as publish and subscribe to insert or 'update data into a stream of data', and to receive the data in that stream, respectively, column 17, line 12-14, e.g. result set). Klein further teaches the instances can produce the application data from the results that are streamed to load queues for a single update to the data store with all the application data, which is to be subsequently accessed from the data store (see col. 13, lines 64 to col. 14, lines 6, column 19, line 17-35, column 15, line 67 to column 16, line 11, 'request type is used for streaming', delete access and update access queries on non-partitioned tables, or if only 'a single partition is accessed by a query', column 14, line 59-64), and wherein the update to the data store is done after each instance of the applications finishes its processing and has streams its application data to the load queues (see abstract, during execution of a select statement that includes an embedded update or delete operation, a table access operator accesses a defined range of rows in a database table, column 11, line 31-36, executed when a table access operator has finished its regular scan of the specified range of rows. The delta scan procedure processes any 'additional rows of data that are ready for processing', and then 'goes to sleep' on the deltascan waiters list 'until more rows of data are ready for processing', column 14, line 59-66, the horizontal partitioning of database tables to queue and publication channels, and uses partitioning for data dependent routing and load distribution).

Therefore, Examiner concludes teaching of Klein in view of Reed teaches each and every elements of claim recites limitations per the arguments presented supra and rejection presented below.

Klein and Reed do not need to disclose anything over and above the invention as claimed in order to render it unpatentable or anticipate. A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the Claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claimed limitations. For the above reasons, it is believed that the rejections should be sustained.

Claim Rejections- 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 8-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al. (US 6453313 B1) ('Klein' hereinafter) and further in view of Reed et al. (US 5862325 A) ('Reed' hereinafter).

With respect to claim 8, Klein teaches a method to manage interactions between applications and a data store (see Figs. 6, 15), comprising:

receiving a query for a data store and an identifier for an application, wherein the application when executed seeks to process results returned from and produced by executing the query and seeks to update the data store with application data, wherein the application data is produced in response to the application processing the results of the query (The fan out operator sends this request 'query' to each table partition, and receives in response all records that satisfy the cursor. The request is non-blocking because the fan out operator does not want or need to receive records added 'update' to the table partition after the request is made. This type of request is used for streaming, read only access (i.e., for streaming operators that do not delete or update tuples). This type of request is sent by the fan out operator to all of the partition scan operators so as to automatically retrieve rows as they are inserted or updated in the table. The delete and update features of the present invention provide a destructive read capability and a "read modify write" capability in conjunction with streaming access to a database table. This allows queuing services to be provided by a relational database system while preserving the ability of the DBMS to perform other relational operators on the result set returned. The result sets created by the delete and update access operations of the present invention can be joined with the result sets of other table access operators, which enables efficient data processing through the use of delete and/or update operations embedded in a query, see col.15, lines 48-60, column 16, lines 66 to column 17, lines 6, column 18, line 12-15, Figs.15, 18, Klein),

concurrently executing initiating multiple instances of the application associated with the identifier on multiple processing nodes within a network to achieve parallel processing for the multiple instances of the application (tables in the database are partitioned, with various partitions being stored on different nodes of the relational database system. Such partitioning is often used for extremely large tables. Various tables within a database are stored on different nodes of the system. Such distributed storage facilitates efficient, parallel 'concurrent' processing of queries, by distributing both the disk I/O and computational burden over multiple nodes. The "application process" represents the process or processes that execute not only the application program, but also the portions of the execution tree above the leaf nodes. The leaf nodes are executed by disk processes in each of the nodes of the transaction processing system. While one disk process for each node, the number of disk processes per node may vary from one implementation to another. A separate disk process may be used for each logical disk volume. Destructive reads are sometimes used to ensure that an item is processed exactly once. For instance, several "credit evaluation" processes might be assigned the job of reading and processing credit applications. Each such process could use a destructive read (i.e., delete operation with result set) to read a next credit application record for processing. The credit evaluation processes work in parallel, without interfering with each other, see col. 5, lines 51-67, column 6, line 57-62, column 14, line 42-58, column 17, line 9-16, Klein).

concurrently processing the query to acquire the results on behalf of the multiple instances of the application (the delete and update features of the present invention provide a destructive read capability and a "read modify write" capability in conjunction with streaming access to a database table. This allows queuing services to be provided by a relational database system while preserving the ability of the DBMS to perform other relational operators on the result set returned. The result sets created by the delete and update access operations of the present invention can be joined with the result sets of other table access operators, column 5, line 51-67, column 16, line 66 to column 17, line 5, column 18, line 12-16, Klein), producing the results that are then streamed to plurality of application queues residing on a plurality of the processing nodes as the results are acquired (data flows between the nodes of the execution tree are handled by the use of a pair of queues, between parent and child nodes. In particular each parent node is coupled to a child node by a request queue and a fetched records queue. The request queue stores requests being conveyed from the parent node to its child node, while the fetched records queue conveys data and return codes (e.g., an end of file or end of scan code) being returned to the parent node in response to the requests (see col. 6, lines 1-10, column 15, line 39-53, Klein).

concurrently providing the results to each of the instances of the application from the one or more application queues so that the instances can produce the application data from the results that are streamed to load queues for a single update to the data store with all the application data, which is to be subsequently accessed from the data store (the request queue and a fetched records queue are used by the transaction processing system to pre-fetch records not yet requested by the application that submitted the query being processed. Each node in the execution tree other than the leaf nodes are automatically configured to request as many records as can be stored in the fetched records queue(s) between it and its child or children nodes, even if such records have not yet been requested by the application. Pre-fetching can improve system performance, by making use of otherwise dormant system resources, and can improve system responsiveness by having data ready for the application before it requests it, unbound pre-fetching must be suppressed when executing an embedded delete or update statement. The application must control how many rows are to be affected by the delete or update operation, and therefore the database management system must only delete or update those records actually requested by the application, SQL compiler includes in the code for any update, delete or insert operator (generically herein called a table access operator) code for generating a before and after image for each modified and new tuple. SQL compiler of the present invention the image generation code includes code for updating one or more fields of the Before Image when the query being compiled includes a 'set on rollback' clause that affects the table being accessed by this operator. When the Before and After Images are passed by the table access operator to the transaction log manager, the before image contains one or more modified fields if the query being executed contained a corresponding "set on rollback" clause, see col. 13, lines 64 to col. 14, lines 6, column 19, line 17-35, column 15, line 67 to column 16, line 11, Klein) and wherein the update to the data store is done after each instance of the applications finishes its processing and has streams its application data to the load queues (see abstract, during execution of a select statement that includes an embedded update or delete operation, a table access operator accesses a defined range of rows in a database table, column 11, line 31-36, executed when a table access operator has finished its regular scan of the specified range of rows. The delta scan procedure processes any 'additional rows of data that are ready for processing', and then 'goes to sleep' on the deltascan waiters list 'until more rows of data are ready for processing', column 14, line 59-66, the horizontal partitioning of database tables to queue and publication channels, and uses partitioning for data dependent routing and load distribution).

Klein does not explicitly teach 'each application queue having different portions of the results'. However, Reed teaches 'each application queue having different portions of the results' (see column 34, line 3-6, these transmissions can be queued using scheduled events to reduce system load, column 45, line 35-37, scheduled objects are communications objects used to represent scheduled events, column 90, line 62-65, maintaining the queue of event instances, column 23, line 40-44, the event class is an abstract class defining the attributes for scheduled events, used to create a queue of events, column 8, line 11-14, results in an updated version is transferred, column 26, 53-56, result of a manual request or it can be transferred directly to the consumer program as a result of automatic event processing, column 47, line 6-8, results in specific pages being assigned to the communications object instance that will transmitted, column 155, line 61-65, transferring at least a portion of the updated information, column 156, claim 92, column 75, line 17-21, 'thread' of message can be passed back and forth, column 75, line 41-49, column 94, line 43-53, see col. 19, line 2-6, line 31-35, col. 94, line 43-53, Reed). Note: Reed teaches data (or object) on all or portion of the data is being transmitted, the transmission result is loaded or scheduled in a queue, while result (or update) have different portion as well.

Klein provides a relational database system that has been extended to perform operations on a continuous stream of tuples (see column 1, line 19-23), while Reed teaches an automated communications system which coordinates the transfer of data, metadata, and instructions between databases in order to control and process communications (column 1, line 11-14). One of ordinary skill in the art at the time of the invention would have been motivated to include the features as taught by Reed to improve the relational database system that has been extended to perform operations on a continuous stream of tuples of Klein for a communications system which allows providers and consumers to easily share access to many common communications services.

Therefore, it would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains to have modified the teaching of Klein by applying the teaching of Reed for transmitted communications object instance can be automatically received, processed, stored and indexed by the consumer program, hence data can be easily searched using consumer program in order to locate specific information or perform certain function, as taught by Reed in column 34, line 49-56.

As to claim 9,

Klein teaches concurrently housing the application data in one or more load queues residing on one or more of the processing nodes (see col. 14, lines 59-63, Klein); and concurrently populating one or more tables on the processing nodes with the application data from the one or more load queues (see col. 14, lines 59-63, Klein).

As to claim 10,

Klein teaches merging the one or more tables into the data store (see col. 3, lines 15-18, Klein).

As to claim 11,

Klein teaches wherein the currently initiating further includes determining a total number of the applications to initiate based on configuration data (see col. 6, lines 1-10, Klein).

As to claim 12,

Klein teaches wherein the currently initiating further includes determining which of a number of the applications that are to be initiated on which of a number of the processing nodes based on the configuration data (see col. 14, lines 10-15, Klein).

As to claim 13,

Klein teaches concurrently synchronizing the application queues and the load queues on the multiple processing nodes when at least some of the processing nodes lack one of the application queues or one of the one or more load queues (see col. 14, lines 59-63, Klein).

As to claim 14,

Klein teaches wherein the concurrently synchronizing further includes establishing socket based communications between the multiple processing nodes with a Transmission ControlProtocol/Internet Protocol (TCP/IP) (see col. 19, lines 30-35, Klein).

Claim 15 have the same subject matter as claim 8 except for the limitations of load queue, system claim, memory device and Kein teaches (see abstract, e.g. system and col. 14, lines 59-63). Therefore, claim 15 is rejected for the same reason as applied to claim 8 hereinabove.

Claims 16 and 17 have the same subject matter as claims 11 and 12 and are rejected for the same reason as applied to claims 11 and 12 hereinabove.

As to claim 18,

Klein teaches wherein each of the applications concurrently processes the results and produces different portions of the application data (see column 5, line 51-67, column 16, line 66 to column 17, line 5, column 18, line 12-16).

As to claim 19,

Klein teaches wherein each of the application queues and each of the load queues concurrently update while the one-or more applications process (see col. 14, lines 59-63).

Claim 20 have the same subject matter as claim 8 except for the limitations of temporary tables, memory device and Kein teaches (see abstract, see column 8, line 43-52, col. 11, lines 11-17, lines 47-55, col. 15, lines 26-35, lines 66 to column 16, line 1-11, and line 58-65 et seq.,). Therefore, claim 20 is rejected for the same reason as applied to claim 8 hereinabove.

As to claim 21,

Klein teaches wherein a merge utility merges the one temporary tables to produce the application data table once each of the multiple instances of the application have finished processing the query results (see col. 13, lines 64 to col. 14, lines 6, column 19, line 17-35, col. 3, lines 15-18, Klein).

As to claim 22,

Klein teaches wherein one or more extract utilities perform a query against the data store in order to acquire the query results, which are concurrently consumed by the multiple instances of the application to produce the application data (see col. 5, lines 51-67, column 6, line 57-62, column 14, line 42-58, column 17, line 9-16, column 11, line 47-50, Klein).

As to claim 23,

Klein teaches wherein each of the one or more extract utilities concurrently populate the query results to application queues (see col. 14, lines 59-63, Klein).

As to claim 24,

Klein teaches wherein each of one or more load utilities concurrently receive portions of the application data from load queues and concurrently populate the portions to the temporary tables. see col. 14, lines 59-63, Klein).

As to claim 25,

Klein teaches wherein the data store is a least one of one or more databases and a data warehouse (col. 15, lines 26-35, lines 66 to column 16, line 1-11, and line 58-65 et seq.).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Kerwin et al. (USP, 6,898,609) teaches all the limitations especially "provides a software method, for network database environments, permitting load balancing, scalability and substantially simultaneous use by client users, comprising the steps of: providing multiple database instances wherein each such instance is substantially identical in data content, database structure, and primary key system; maintaining substantially real time records of status for each such multiple database instance; receiving a database query from at least one end-user application and determining such query to be a transactional query or non-transactional query; directing such database query to at least one selected instance of such multiple database instances upon a determination of such query being a non-transactional query; returning such non-transactional query results to the at least one end-user application; directing such database query to all instances of such multiple database instances upon a determination of such query being a transactional query; controlling such transactional queries to maintain substantial identicalness

among such multiple database instances; propagating such transactional queries to such multiple database instances; returning such query results to the user; recognizing a failure in at least one instance of such multiple database instances, and adjusting to store such transactional query for later propagation; restoring such failed at least one instance of such multiple database instances to substantial identicalness with other such multiple database instances. Moreover, it provides such a method wherein each such non-transactional query is executed upon a randomly selected instance of such multiple database instances. Additionally, it provides such a method wherein the processing of such non-transactional query commands as directed by a plurality of users is substantially simultaneous" see col. 5, lines 1-25, and col. 7, lines 1-16, Kerwin et al.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communication from the examiner should be directed to Daniel A Kuddus whose telephone number is (571) 270-1722. The examiner can normally be reached on Monday to Thursday 8.00 a.m.-5.30 p.m. The examiner can also be reached on alternate Fridays from 8.00 a.m. to 4.30 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or processing is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from the either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Daniel Kuddus

Date: 03/19/09

/Charles Rones/

Supervisory Patent Examiner, Art Unit 2164